A New Digital Image Steganography Approach Based on The Galois Field GF(pm) Using Graph and Automata

Nguyen Huy Truong
School of Applied Mathematics and Informatics, Hanoi University of Science and Technology, Hanoi, Viet Nam
[e-mail: truong.nguyenhuy@hust.edu.vn]

Received October 15, 2018; revised December 21, 2018; accepted March 5, 2019; published September 30, 2019

Abstract

In this paper, we introduce concepts of optimal and near optimal secret data hiding schemes. We present a new digital image steganography approach based on the Galois field $GF(p^m)$ using graph and automata to design the data hiding scheme of the general form $(k, N, \lfloor \log_2 p^{mn} \rfloor)$ for binary, gray and palette images with the given assumptions, where k, m, n, N are positive integers and p is prime, show the sufficient conditions for the existence and prove the existence of some optimal and near optimal secret data hiding schemes. These results are derived from the concept of the maximal secret data ratio of embedded bits, the module approach and the fastest optimal parity assignment method proposed by Huy et al. in 2011 and 2013. An application of the schemes to the process of hiding a finite sequence of secret data in an image is also considered. Security analyses and experimental results confirm that our approach can create steganographic schemes which achieve high efficiency in embedding capacity, visual quality, speed as well as security, which are key properties of steganography.

Keywords: Cryptography, Steganography, Module, Galois field, Graph, Automata.

http://doi.org/10.3837/tiis.2019.09.025 **ISSN : 1976-7277**

1. Introduction

In the modern life, when the use of computer and Internet is more and more essential, digital data can be copied as well as accessed illegally. As a result, data security becomes increasingly important. There are two popular ways to provide security, which are cryptography and data hiding [1, 3, 4, 8, 17, 23]. Cryptography is used to encrypt data in order to make the data unreadable by a third party [3]. Data hiding is used to embed data in digital media. Based on the purpose of the application, data hiding is generally divided into steganography that hide the existence of data to protect the embedded data and watermarking that protect the copyright ownership and authentication of the digital media carrying the embedded data. Depend on the type of digital media there are many types of steganography, for example image, audio and video steganography [2, 3, 8, 16, 17, 21, 22, 27]. This work only focuses on steganography in digital images in spatial domain. Then the steganography is achieved by changing colors of some pixels directly in the image [17, 22].

The steganography studies the steganographic schemes, where each scheme consists of an embedding function and extracting function. The embedding function shows how to embed secret data in the digital image (called the cover image) and the extraction function describes how to extract the data from the digital image carrying the embedded data (called the stego image) [11, 24].

In the steganography, a few main factors must be taken in consideration when we design a new secret data hiding scheme, which are embedding capacity of the cover image, quality of stego image and security. However, as well known, embedding capacity of the cover image and quality of its stego image are irreconcilable conflict. A balance achieved of the two factors can be done according to different application requirements. In addition to the three main factors, speed of the embedding and extracting functions also plays an important role in steganographic schemes. It is considered as a last constraint to determine efficiency of schemes [11, 15, 18, 20, 24, 29].

The simplest and most popular spatial domain image steganography method is the least significant bit (LSB) substitution (called LSB based method). For 24-bit RGB and 8-bit gray images, in this method the data is embedded in the cover image by changing the least significant bits of the image directly, therefore it becomes vulnerable to security attacks [7, 17, 19, 21, 22, 28, 29]. EZ Stego method for palette images is similar to the commonly used LSB based method. However, this method does not guarantee quality of stego images [9, 10, 28]. To alleviate this problem, in 1999, Fridrich proposed a new method based on the parity bits of color indexes of pixels in palette cover images, called the parity assignment (PA) method. Then EZ Stego method can be considered as an example of PA method [9, 14]. In 2000, Fridrich et al. improved the method by investigating the problem of optimal parity assignment for the palette and this version is called the optimal parity assignment (OPA) method [10]. To easily control quality of stego images, Huy et al. introduced another OPA method, called the fastest optimal parity assignment (FOPA) method, in 2013 [14]. Unlike the color and gray images, each pixel in binary images only requires one bit to represent color values (black and white), therefore, modifying pixels can be easily detected. So, binary image steganography is a more difficult and challenging problem. For binary images, block based method is usually used to maintain quality of stego images. In this method, the cover and stego images are partitioned into individual image blocks of the same size, embedding and extracting secret data are based on the characteristic values calculated for the blocks. WL (Wu et al., 1998), PCT (Pan et al., 2000), modified PCT (Tseng et al., 2001), CTL

(Chang et al., 2005) schemes are all well known and block based for binary images [6, 7, 12, 21, 26].

Given a positive integer q_{color} which is the number of different ways to change the color of each pixel in an arbitrary image block, used the concept of the maximal secret data ratio of embedded bits proposed by Huy et al. in 2011 [13], we introduce concepts of optimal and near optimal secret data hiding schemes. Actually, the optimality of steganographic schemes has been considered in [10, 11]. However, the authors used the time complexity of embedding and extracting functions, or the concept of optimal parity assignment that minimizes the energy of the parity assignment for the color palette to determine whether a steganographic scheme is optimal.

By the block based method, call a secret data hiding scheme a data hiding scheme (k,N,r), where k,N,r are positive integers, if the embedding function can embed r bits of secret data in each image block of N pixels by changing colors of at most k pixels in the image block. Our work is concerned with the problem of designing optimal or near optimal data hiding schemes (k,N,r), for digital images (binary, gray and palette images).

In this paper, based on the module approach and the fastest optimal parity assignment (FOPA) method proposed by Huy et al. in 2011 and 2013 [13, 14], we introduce a new approach based on the Galois field using graph and automata in order to solve the problem. For the purpose of our research, the proposed schemes consist of the optimal data hiding scheme $(1,2^n-1,n)$ for binary, gray and palette images with $q_{color}=1$, where n is a positive integer, the near optimal data hiding scheme $(2,9,8)$ for gray and palette images with $q_{color}=3$ and the optimal data hiding scheme $(1,5,4)$ for gray and palette images with $q_{color}=3$. Security analyses show that an application of these schemes to the process of hiding a finite sequence of secret data in an image can avoid detection from brute-force attacks.

The experimental results reveal that the efficiency in embedding capacity and visual quality of the near optimal data hiding scheme $(2,9,8)$ for gray images with $q_{color}=3$ is indeed better than the efficiency of the HCIH scheme [29]. The embedding and extracting time of our approach are faster than that of the Chang et al.'s approach [7]. For the near optimal data hiding scheme $(2,9,8)$ for palette images with $q_{color}=3$ and the optimal data hiding scheme $(1,2^n-1,n)$ for palette images with $q_{color}=1$, we can choose suitable values of *ER* to achieve acceptable quality of the stego images.

The rest of the paper is organized as follows. In Section 2, we recall the notation *MSDR* [13], give some new concepts and state our digital image steganography problem. Section 3 consists of three Subsections 3.1, 3.2 and 3.3. In Subsection 3.1, we introduce mathematical basis based on the Galois field $GF(p^m)$ for the digital image steganography problem, where p is prime and m is a positive integer (Propositions 3.4, 3.11 and Theorem 3.10). In Subsection 3.2, firstly, we propose a digital image steganography approach based on the Galois field $GF(p^m)$ using graph and automata to design the data hiding scheme of the general form $(k,N,\lfloor \log_2 p^{mn} \rfloor)$ for the given assumptions, where k,m,n,N are positive integers and p is prime (Theorem 3.20 and Security analysis (3.12)). Secondly, we give the sufficient conditions for the existence of the optimal data hiding schemes

$(1,\frac{p^{mn}-1}{p^m-1},\lfloor \log_2 p^{mn} \rfloor)$ and $(2,\left\lceil \frac{\frac{p^m-3}{2}+\sqrt{\frac{(p^m-3)^2}{4}+2(2^{\lfloor \log_2 p^{mn} \rfloor}-1)}}{p^m-1} \right\rceil,\lfloor \log_2 p^{mn} \rfloor)$ with $q_{color} = p^m - 1$ (Theorems 3.21 and 3.23). Thirdly, we show that there exists the optimal data hiding scheme $(1, 2^n - 1, n)$ for binary, gray and palette images with $q_{color} = 1$, where n is a positive integer (Proposition 3.22). At the end of the Subsection 3.2, we consider the way to apply the data hiding scheme $(k, N, \lfloor \log_2 p^{mn} \rfloor)$ to the process of hiding a finite sequence of secret data of length $\lfloor log_2 p^{mn} \rfloor$ bits in an image (Proposition 3.24 and Security analysis (3.27)). In Subsection 3.3, we prove that there exist the near optimal data hiding scheme $(2,9,8)$ (Theorem 3.27 and Security analyses (3.45), (3.46)) and the optimal data hiding scheme $(1,5,4)$ (Corollary 3.28 and Security analyses (3.47), (3.48)) for gray and palette images with $q_{color} = 3$. Section 4 shows experimental results in order to evaluate the efficiency of our proposed data hiding schemes and approach. Lastly, we draw some conclusions from our approach and experimental results in Section 5.

2. The Digital Image Steganography Problem

In this section, we recall the notation *MSDR* [13], give some new concepts and state our digital image steganography problem.

Definition 2.1. A block based secure data hiding scheme in digital images (for short, called a data hiding scheme) is a five tuple $(\mathcal{I}, \mathcal{M}, \mathcal{K}, Em, Ex)$, where the following conditions are satisfied.

1. $\mathcal{I}$ is a set of all image blocks with the same size and image format,
2. $\mathcal{M}$ is a finite set of secret elements,
3. $\mathcal{K}$ is a finite set of secret keys,
4. Em is an embedding function to embed a secret element in a image block, $Em : \mathcal{I} \times \mathcal{M} \times \mathcal{K} \to \mathcal{I}$,
5. Ex is an extracting function to extract a embedded secret element from a image block, $Ex : \mathcal{I} \times \mathcal{K} \to \mathcal{M}$,
6. $Ex(Em(I, M, K), K) = M, \forall (I, M, K) \in \mathcal{I} \times \mathcal{M} \times \mathcal{K}$.

Definition 2.2. A data hiding scheme $(\mathcal{I}, \mathcal{M}, \mathcal{K}, Em, Ex)$, is called a data hiding scheme (k, N, r), where k, N, r are positive integers, if each image block in $\mathcal{I}$ has N pixels and the embedding function Em can embed r bits of secret data in an arbitrary image block by changing colors of at most k pixels in the image block.

Definition 2.3 ([13])**.** $MSDR_k(N)$ is the largest number of embedded bits of secret data in an image block of N pixels by changing colors of at most k pixels in the image block, where k, N are positive integers.

Note that *MSDR* is the abbreviation for 'Maximal Secret Data Ratio' [13].

Given a positive integer q_{color}, we call q_{color} the number of different ways to change the color of each pixel in an arbitrary image block of N pixels. According to [13]

$$MSDR_k(N) = \lfloor log_2(1 + q_{color}C_N^1 + q_{color}^2 C_N^2 + \cdots + q_{color}^k C_N^k) \rfloor. \quad (2.1)$$

Definition 2.4. For a given q_{color}, a data hiding scheme (k, N, r) is called an optimal data hiding scheme if $r = MSDR_k(N)$ and $\nexists N'$, $N' < N$, $r = MSDR_k(N')$. Then N is denoted by $N_{optimum}$.

Definition 2.5. For a given q_{color}, a data hiding scheme (k, N, r) is called a near optimal data hiding scheme if $r = MSDR_k(N)$ and $N > N_{optimum}$.

Our digital image steganography problem. Design optimal or near optimal data hiding schemes (k, N, r) for digital images (binary, gray and palette images).

3. Main Results

In this section, we introduce mathematical basis based on the Galois field for the digital image steganography problem (Subsection 3.1), propose a digital image steganography approach based on the Galois field using graph and automata to design the data hiding scheme of the general form $(k, N, \lfloor \log_2 p^{mn} \rfloor)$ for the given assumptions, where k, m, n, N are positive integers and p is prime (Subsection 3.2), and show the sufficient conditions for the existence or prove the existence of some optimal and near optimal data hiding schemes (Subsections 3.2 and 3.3). Security analyses and an application of these data hiding schemes to the process of hiding a finite sequence of secret data in an image are considered in Subsections 3.2 and 3.3.

3.1 Mathematical Basis based on The Galois Field for Digital Image Steganography

In this subsection, we construct mathematical basis based on the Galois field $GF(p^m)$ for the digital image steganography problem, where p is prime and m is a positive integer (Propositions 3.4, 3.11 and Theorem 3.10).

Now, we consider the Galois field $GF(p^m)$ to be constructed from the polynomial ring $Z_p[x]$, where p is prime and m is a positive integer [25]. Let $GF^n(p^m) = \{(x_1, x_2, \ldots, x_n) \mid x_i \in GF(p^m), \forall i = \overline{1,n}\}$, where n is a positive integer, with two operations of vector addition + and scalar multiplication · are defined as follows.

$$x + y = (x_1 + y_1, x_2 + y_2, \ldots, x_n + y_n),$$
$$ax = (ax_1, ax_2, \ldots, ax_n), a \in GF(p^m),$$

where $x, y \in GF^n(p^m)$ and $x = (x_1, x_2, \ldots, x_n), y = (y_1, y_2, \ldots, y_n)$. We remember that $(GF^n(p^m), +, \cdot)$ is a vector space over the field $GF(p^m)$ [5].

Definition 3.1. The class of an element $x \in GF^n(p^m)$, denoted by [x], is given by

$$[x] = \{ax \mid a \in GF(p^m) \setminus \{0\}\}.$$

Given a class [x], x is referred to as the representative of [x]. For simplicity, denote the class [0] by 0.

Lemma 3.2. *For all* $x, y \in GF^n(p^m)$, $[x] \cap [y] = \varnothing$ *or* $[x] = [y]$.

Proof. Suppose $[x]\cap[y]\neq\varnothing$, then $\exists z\in[x]\cap[y]$. By Definition 3.1, $z=ax=by$. Since $a\in GF(p^m)\setminus\{0\}$, $x=a^{-1}by$. Thus $x\in[y]$ and therefore $[x]\subset[y]$. Similarly, $[y]\subset[x]$ and hence $[x]=[y]$.

Propostion 3.3. *The set of all* classes *forms a partition of the set* $GF^n(p^m)$.

Proof. For $\forall x\in GF^n(p^m)$, then $x\in[x]$ by Definition 3.1. Thus the union of all classes is $GF^n(p^m)$. By Lemma 3.2, any two distinct classes are disjoint. The proof is complete.

Denote the set of all classes by $[GF^n(p^m)]$. This can be represented by $[GF^n(p^m)]=\{[x]\mid x\in GF^n(p^m)\}$. The number of elements of a set S is denoted by $|S|$.

Propostion 3.4. $|[GF^n(p^m)]\setminus\{0\}|=\dfrac{p^{mn}-1}{p^m-1}$.

Proof. Evidently, $|GF^n(p^m)\setminus\{0\}|=p^{mn}-1$. Consider $y,y'\in[x],y\neq y',[x]\in[GF^n(p^m)]\setminus\{0\}$, then $y=ax,y'=bx$ for $a,b\in GF(p^m)\setminus\{0\}$. Since $y\neq y',x\neq 0$, then $a\neq b$. Clearly, $|GF(p^m)\setminus\{0\}|=p^m-1$ (see [25]). Since $x\neq 0$, then $|[x]|=p^m-1$. By Proposition 3.3, $|[GF^n(p^m)]\setminus\{0\}|=\dfrac{p^{mn}-1}{p^m-1}$.

Definition 3.5. Suppose $S\subset[GF^n(p^m)]\setminus\{0\}$, Then S is called a *k-[Generators]* for the set $[GF^n(p^m)]$, where k is a positive integer, if $\forall[v]\in[GF^n(p^m)]\setminus\{0\},[v]\in\{[\sum_{i=1}^{t}a_iv_i]\mid a_i\in GF(p^m)\setminus\{0\},[v_i]\in S,i=\overline{1,t},t\leq k\}$.

Propostion 3.6. *If S is a k-[Generators] for the set* $[GF^n(p^m)]$, *where k is a positive integer, then S does not depend on the choice of representatives of classes.*

Proof. It may be more convenient to prove, we assume that $[v_i']=[v_i]$ for all i, set

$$A=\{[\sum_{i=1}^{t}a_iv_i]\mid a_i\in GF(p^m)\setminus\{0\},[v_i]\in S,i=\overline{1,t},t\leq k\},$$

$$B=\{[\sum_{i=1}^{t}a_i'v_i']\mid a_i'\in GF(p^m)\setminus\{0\},[v_i']\in S,i=\overline{1,t},t\leq k\}.$$

To prove that S does not depend on the choice of representatives of classes, it suffices to show that $A=B$. By the hypothesis $[v_i']=[v_i]$, then $v_i=b_iv_i'$. Suppose $[x]\in A$, then $x=\alpha(\sum_{i=1}^{t}a_iv_i)=\alpha(\sum_{i=1}^{t}a_ib_iv_i')$. Clearly, $a_ib_i\neq 0$ by the definition of the class, then $[x]\in B$. Conversely, since $b_i\neq 0$, then $\exists b_i^{-1}$, thus $v_i'=b_i^{-1}v_i$. Similarly, $B\subset A$. So, $A=B$.

Definition 3.7. Let V be a vector space over a field K, $S\subset V$. Then S is called a *k-Generators* for V, where k is a positive integer, if the two following conditions are satisfied.

a) $\forall v,v'\in S,\nexists a\in K,v'=av$,

b) $\forall v\in V\setminus\{0\},\exists t,t\leq k,v_1,v_2,\ldots,v_t\in S,a_1,a_2,\ldots,a_t\in K\setminus\{0\},v=\sum_{i=1}^{t}a_iv_i$.

Lemma 3.8. *Let* $S=\{v_1,v_2,\ldots,v_t\}$ *be a k-Generators for the vector space* $GF^n(p^m)$. *Then* $S'=\{[v_1],[v_2],\ldots,[v_t]\}$ *is a k-[Generators] for the set* $[GF^n(p^m)]$.

Proof. Since S is a *k-Generators* for $GF^n(p^m)$, then $\forall v,v'\in S,\nexists a\in GF(p^m),v'=av$. By Proposition 3.3 and Definition 3.1, $[v_i]\neq[v_i']$ and $[v_i]\neq 0,\forall v_i\in S,1\leq i\leq t$. For all

$[u] \in [GF^n(p^m)] \setminus \{0\}$, then $u = \sum_{i=1}^{k'} a_i v_{j_i}$, $k' \le k, v_{j_i} \in S, a_i \in GF(p^m) \setminus \{0\}, i = \overline{1,k'}$. Thus $[u] = [\sum_{i=1}^{k'} a_i v_{j_i}]$ and hence $[u] \in \{[\sum_{i=1}^{k'} a_i v_{j_i}] \mid a_i \in GF(p^m) \setminus \{0\}, [v_{j_i}] \in S', i = \overline{1,k'}, k' \le k\}$. The proof is complete. □

Lemma 3.9. *Let* $S' = \{[v_1],[v_2],\ldots,[v_t]\}$ *be a k-[Generators] for the set* $[GF^n(p^m)]$. *Then* $S = \{v_1, v_2, \ldots, v_t\}$ *is a k-Generators for the vector space* $GF^n(p^m)$.

Proof. For all $v \in GF^n(p^m) \setminus \{0\}$, then $[v] \ne 0, [v] \in \{[\sum_{i=1}^{k'} a_i v_{j_i}] \mid a_i \in GF(p^m) \setminus \{0\}, [v_{j_i}] \in S', i = \overline{1,k'}, k' \le k\}$. Thus $\exists v = \alpha(\sum_{i=1}^{k'} a_i v_{j_i}) = \sum_{i=1}^{k'} (\alpha a_i) v_{j_i}, \alpha, a_i \in GF(p^m) \setminus \{0\}, [v_{j_i}] \in S', v_{j_i} \in S, i = \overline{1,k'}, k' \le k$. For all $[v],[v'] \in S'$, then $\nexists a \in GF(p^m), v' = av$ by Proposition 3.3. It means that $\forall v, v' \in S, \nexists a \in GF(p^m), v' = av$. The proof is complete.

Theorem 3.10. *There exists S to be a k-Generators for the vector space* $GF^n(p^m)$ *with* $|S| = N$ *if and only if there exists* S' *to be a k-[Generators] for the set* $[GF^n(p^m)]$ *with* $|S'| = N$.

Proof. This is deduced immediately from Lemmas 3.8 and 3.9.

Propostion 3.11. *Let c be the number of k-[Generators] of N elements for the set* $[GF^n(p^m)]$. *Then the number of k-Generators of N elements for the vector space* $GF^n(p^m)$ *is* $c(p^m - 1)^N$.

Proof. Suppose S' is a *k-[Generators]* for $[GF^n(p^m)]$ with $|S'| = N$. Since S' does not depend on the choice of representatives of classes by Proposition 3.6, the number of ways to change representatives of all classes in S' is $c(p^m - 1)^N$. By the hypothesis, the number of *k-[Generators]* of *N* elements for the set $[GF^n(p^m)]$ is *c*, then the number of *k-Generators* of *N* elements for the vector space $GF^n(p^m)$ is $c(p^m - 1)^N$ by Lemma 3.9 and Theorem 3.10.

3.2 A New Digital Image Steganography Approach Based on The Galois Field GF(p^m) Using Graph and Automata

In this subsection, firstly, we introduce a digital image steganography approach based on the Galois field $GF(p^m)$ using graph and automata to design the data hiding scheme of the general form $(k, N, \lfloor log_2 p^{mn} \rfloor)$ for the given assumptions, where k, m, n, N are positive integers and *p* is prime (Theorem 3.20 and Security analysis (3.12)). Secondly, we give the sufficient conditions for the existence of the optimal data hiding schemes $(1, \frac{p^{mn}-1}{p^m-1}, \lfloor log_2 p^{mn} \rfloor)$ and $(2, \left\lceil \frac{\frac{p^m-3}{2} + \sqrt{\frac{(p^m-3)^2}{4} + 2(2^{\lfloor log_2 p^{mn} \rfloor} - 1)}}{p^m - 1} \right\rceil, \lfloor log_2 p^{mn} \rfloor)$ with $q_{color} = p^m - 1$ (Theorems 3.21 and 3.23). Thirdly, we show that there exists the optimal data hiding scheme $(1, 2^n - 1, n)$ for binary, gray and palette images with $q_{color} = 1$, where *n* is a positive integer (Proposition 3.22). And finally, we present the way to apply the data hiding

scheme $(k, N, \lfloor log_2 p^{mn} \rfloor)$ to the process of hiding a finite sequence of secret data of length $\lfloor log_2 p^{mn} \rfloor$ bits in an image (Proposition 3.24 and Security analysis (3.27)).

Let $\mathcal{I}$ be a set of all image blocks with the same size and image format and assume that each image block in $\mathcal{I}$ has *N* pixels, where *N* is a positive integer. For simplicity, we can consider the structure of an arbitrary image block *I* in $\mathcal{I}$ to be represented by $I = \{I_1, I_2, ..., I_N\}$, where I_i is a color value for binary and gray images or color index in the palette for palette images of the i^{th} pixel in *I*, $\forall i = \overline{1, N}$. Consider *C* to be a set of all color values or indexes of all pixels belonged to $\mathcal{I}$.

Let $\mathcal{M}$ be a finite set of secret elements and set $\mathcal{M} = GF^n(p^m)$.

Let $\mathcal{K}$ be a finite set of secret keys. For all $K \in \mathcal{K}$, we also assume that the structure of the key *K* is the same as the structure of the image block *I*. So, we can write $K = \{K_1, K_2, ..., K_N\}$ for $K_i \in GF(p^m), \forall i = \overline{1, N}$.

Assume that we can find a *k-Generators S* for $GF^n(p^m)$ with $|S| = N$ and $S = \{v_1, v_2, \ldots, v_N\}$.

Definition 3.12. A weighted directed graph $G = (V,E)$ is called a flip graph over the Galois field $GF(p^m)$ (for short, called a flip graph) if the two following conditions are satisfied.

1. $V = C$ and for all $v \in V$, the vertex *v* is assigned a weight by a function *Val* such that $Val(v) \in GF(p^m)$.
2. For $\forall c_p \in V, \forall a \in GF(p^m) \setminus \{0\}, \exists!(c_p, c_{p'}) \in E$ and the arc $(c_p, c_{p'})$ is assigned the weight *a* such that $Val(c_{p'}) = Val(c_p) + a$ (on $GF(p^m)$).

Given a flip graph *G*, we denote by $Adjacent(c_p,a)$ the vertex adjacent to c_p, where the weight *a* is assigned to the arc $(c_p, Adjacent(c_p, a))$.

Assume that we can build a flip graph $G = (V,E)$.

From the way to determine the edge set *E* in Definition 3.12, we assume that

$$|C| \geq p^m \text{ and } q_{color} = p^m - 1. \tag{3.1}$$

Definition 3.13. Let $\Sigma_1 = \{1, 2, \ldots, N\} \times C, q \in GF^n(p^m), (i, c_p) \in \Sigma_1$. Then δ_1 is a function $\delta_1 : GF^n(p^m) \times \Sigma_1 \to GF^n(p^m)$ defined by $\delta_1(q, (i, c_p)) = q + Val(c_p)v_i$ $(on\ GF^n(p^m))$.

Definition 3.14. Let $\Sigma_2 = GF^n(p^m), \mathcal{N} = \{1, 2, \ldots, N\}, 2^{\mathcal{N} \times GF(p^m) \setminus \{0\}}$ be the set of all subsets of the set $\mathcal{N} \times GF(p^m) \setminus \{0\}$. Then δ_2 is a function $\delta_2 : GF^n(p^m) \times \Sigma_2 \to 2^{\mathcal{N} \times GF(p^m) \setminus \{0\}}$ defined by

$$\delta_2(q,v) = \begin{cases} \{(i_t, a_t) \mid 1 \leq i_t \leq N, t = \overline{1, k'}, k' \leq k, v_{i_t} \in S, a_t \in GF(p^m) \setminus \{0\}, v + (-q) = \sum_{t=1}^{k'} a_t v_{i_t} \ (on\ GF^n(p^m))\} & \text{if } v \neq q, \\ \varnothing & \text{otherwise.} \end{cases}$$

Remark 3.15. For the case $v \neq q$, then $v + (-q) \neq 0$. Since *S* is a *k-Generators* for $GF^n(p^m), |S| = N, S = \{v_1, v_2, \ldots, v_N\}$, thus $\exists k', k' \leq k, \exists v_{i_t} \in S, 1 \leq i_t \leq N, \exists a_t \in GF(p^m) \setminus \{0\}, t = \overline{1, k'}, v + (-q) = \sum_{t=1}^{k'} a_t v_{i_t}$ $(on\ GF^n(p^m))$. So, δ_2 given in Definition 3.14 is a function.

Definition 3.16. Let $I \in \mathcal{I}, M \in \mathcal{M}$ and $K \in \mathcal{K}$. The automaton $A(I,M,K)$ is a five tuple
$(\Sigma, Q, q_0, \delta, T)$, where

- The alphabet $\Sigma = C \cup \Sigma_2$;
- The set of states $Q = \{q_i, i = \overline{0, N+1} \mid q_0 = \sum_{i=1}^{N} K_i v_i, q_i = \delta_1(q_{i-1}, (j, I_i)), \forall i = \overline{1, N}, q_{N+1} = \delta_2(q_N, M)\}$,
- The initial state q_0;
- The set of final states $T = \{q_{N+1}\}$;
- The transition function $\delta : Q \times \Sigma \to Q$, $\delta(q_{i-1}, I_i) = q_i, \forall i = \overline{1, N}, \delta(q_N, M) = q_{N+1}$.

Remark 3.17. The set of states Q and the transition function δ given in Definition 3.16 are completely determined based on the functions δ_1, δ_2 and it follows that the automaton $A(I,M,K)$ is constructed accurately in Definition 3.16.

Let an image block $I \in \mathcal{I}$, a secret element $M \in \mathcal{M}$, a key $K \in \mathcal{K}$. By using the automaton $A(I,M,K)$ and the flip graph G, two functions Em and Ex in the data hiding scheme $(\mathcal{I}, \mathcal{M}, \mathcal{K}, Em, Ex)$, are designed as follows.

The function Em (embedding M in I):

$q = q_0$; (3.2)

For $i = 1$ to N Do $q = \delta(q, I_i)$; (3.3)

$q = \delta(q, M)$; (3.4)

For each (i_t, a_t) in q Do $I_{i_t} = \text{Adjacent}(I_{i_t}, a_t)$; (3.5)

$I' = I$; (3.6)

Remark 3.18. Consider $I' = Em(I,M,K)$, by (3.5), Em only changes colors of $|q|$ pixels in I based on the flip graph G, then $I' \in \mathcal{I}$. So, Em designed holds Definition 2.1.

The function Ex (Extracting M from I'):

$q = q_0$; (3.7)

For $i = 1$ to N Do $q = \delta(q, I'_i)$; (3.8)

$M = q$; (3.9)

From Defintion 2.1, the correctness of the data hiding scheme $(\mathcal{I}, \mathcal{M}, \mathcal{K}, Em, Ex)$, is confirmed by the following proposition.

Propostion 3.19. $\forall (I,M,K) \in \mathcal{I} \times \mathcal{M} \times \mathcal{K}, Ex(Em(I,M,K),K) = M$.

Proof. Set $M' = Ex(I',K)$. By Definitions 3.13 and 3.16, $M' = \sum_{i=1}^{N} (Val(I'_i) + K_i) v_i$ (3.9). After implementing (3.3) $q = \sum_{i=1}^{N} (Val(I_i) + K_i) v_i$. By Definitions 3.14 and 3.16, after implementing (3.4) we consider two cases of q:

If $q = \varnothing$, then (3.5) is not implemented and hence I is not changed. Thus $I' \equiv I$ and therefore $M' = M$.

Otherwise $M + (-q) = \sum_{t=1}^{k'} a_t v_{i_t}, q$ is computed by (3.3), $1 \le i_t \le N, t = \overline{1, k'}, k' \le k, v_{i_t} \in S, a_t \in GF(p^m) \setminus \{0\}$, then

$$M = q + \sum_{t=1}^{k'} a_t v_{i_t} = \sum_{i=1}^{N} (Val(I_i) + K_i) v_i + \sum_{t=1}^{k'} a_t v_{i_t} = \sum_{1 \le i \le N, i \ne i_t, t = \overline{1,k'}} (Val(I_i) + K_i) v_i + \sum_{t=1}^{k'} ((Val(I_{i_t}) + a_t) + K_{i_t}) v_{i_t}.$$

I is changed in positions $i_t, t = \overline{1,k'}$ by (3.5), I_{i_t} is changed to I'_{i_t}, $Val(I'_{i_t}) = Val(I_{i_t}) + a_t$ by the flip graph G.

$$M' = \sum_{i=1}^{N}(Val(I'_i)+K_i)v_i = \sum_{1\le i\le N, i\ne i_t, t=\overline{1,k'}}(Val(I_i)+K_i)v_i + \sum_{t=1}^{k'}(Val(I'_{i_t})+K_{i_t})v_{i_t} = \sum_{1\le i\le N, i\ne i_t, t=\overline{1,k'}}(Val(I_i)+K_i)v_i + \sum_{t=1}^{k'}((Val(I_{i_t})+a_t)+K_{i_t})v_{i_t} = M.$$

Theorem 3.20. *Suppose that find a k-Generators S for the vector space $GF^n(p^m)$ and build a flip graph G. Then there exists the data hiding scheme $(k, N, \lfloor log_2 p^{mn} \rfloor)$, where N = |S|.*

Proof. For the assumption that find a *k-Generators S* for $GF^n(p^m)$, $|S| = N$ and build a flip graph *G*, we offer the way to construct the data hiding scheme $(\mathcal{I}, \mathcal{M}, \mathcal{K}, Em, Ex)$ based on the Galois field $GF(p^m)$ by using the flip graph *G* and the automaton $A(I, M, K)$. *Em* changes colors of at most *k* pixels *I* to embed *M* in *I*, $\forall I \in \mathcal{I}, \forall M \in \mathcal{M}$ by Definition 3.14 and Statement (3.5).

Consider *B* to be the set of all secret data of length *r* bits, then $|B| = 2^r$. $|\mathcal{M}| = p^{mn}$ by $\mathcal{M} = GF^n(p^m)$. Suppose that we can construct an injective function f, $f: B \to \mathcal{M}$. Then we can use the *Em* to embed $b \in B$ in *I* as follows.

$$M = f(b); \quad (3.10)$$
$$I' = Em(I, M, K);$$

Since *f* is injective by our supposition, after extracting *M* from I' by *Ex*, the secret data *b* will be determined accurately based on *f*.

Since *B* and *M* are finite sets, thus to exist the injective function *f*, we let $|B| \le |\mathcal{M}|$, it means $2^r \le p^{mn}$, then $r \le log_2 p^{mn}$, choose $r = \lfloor log_2 p^{mn} \rfloor$. So, for $r = \lfloor log_2 p^{mn} \rfloor$, the *r* bits of the secret data *b* can be embedded in *I*. By Definition 2.2, the data hiding scheme $(\mathcal{I}, \mathcal{M}, \mathcal{K}, Em, Ex)$ is a data hiding scheme $(k, N, \lfloor log_2 p^{mn} \rfloor)$. So, we say that the data hiding scheme $(k, N, \lfloor log_2 p^{mn} \rfloor)$ exists.

Security analysis of the data hiding scheme proposed $(k, N, \lfloor log_2 p^{mn} \rfloor)$: Assume that we publish parameters *k,N, Em, Ex*, the vector space $GF^n(p^m)$ and the flip graph *G* in the data hiding scheme $(k, N, \lfloor log_2 p^{mn} \rfloor)$. The secret element *M* is extracted from I' by the extracting function *Ex* as follows

$$M = Ex(I', K),$$

from Definitions 3.13 and 3.16 and by (3.9):

$$M = \sum_{i=1}^{N}(Val(I'_i)+K_i)v_i. \quad (3.11)$$

By (3.11), to extract accurately *M*, we need to know the *k-Generators S* for $GF^n(p^m)$ and the key *K*. Since the number of the *k-Generators* found is $c(p^m-1)^N$ by Proposition 3.11, then the number of choices for the *k-Generators S* is $c(p^m-1)^N N!$ (note that the order of elements in *S* also affects the formula (3.11)). The number of choices for the key *K* is p^{mN} because $K \in \mathcal{K}$. Consider *GF* to be an arbitrary subset of $2^{\lfloor log_2 p^{mn} \rfloor}$ elements of the set $GF^n(p^m)$, *B* to be the set of all secret data of length $\lfloor log_2 p^{mn} \rfloor$ bits, it means $B = \{0,1,\ldots,2^{\lfloor log_2 p^{mn} \rfloor}-1\}$ in the decimal system. Then there exists a bijective function f, $f: B \to GF$. By (3.10), to decrypt the secret element *M* to the secret data *b*, we need to know *f*. The number of choices for the

bijective function f is $C_{p^{mn}}^{2^{\lfloor log_2 p^{mn} \rfloor}} 2^{\lfloor log_2 p^{mn} \rfloor}!$. Then for a brute force attack, an attacker has to try every possible combination of S, K and f in the given data hiding scheme. The number of combinations of S, K and f is

$$c(p^m-1)^N N! p^{mN} C_{p^{mn}}^{2^{\lfloor log_2 p^{mn} \rfloor}} 2^{\lfloor log_2 p^{mn} \rfloor}!. \quad (3.12)$$

Theorem 3.21. *Suppose that build a flip graph G. Then there exists the optimal data hiding scheme* $(1, \frac{p^{mn}-1}{p^m-1}, \lfloor log_2 p^{mn} \rfloor)$ *for* $q_{color} = p^m - 1$.

Proof. Set $S' = [GF^n(p^m)] \setminus \{0\}$, then S' is a *1-[Generators]* for $[GF^n(p^m)]$ by Definition 3.5. Consider $[v] \in S'$, then $S' \setminus \{[v]\}$ is not a *1-[Generators]* for $[GF^n(p^m)]$ because $[v] \notin \{[av'] \mid a \in GF(p^m), [v'] \in S' \setminus \{[v]\}\}$ by Proposition 3.3). Therefore

$$S' \text{ is the unique } 1\text{-}[Generators] \text{ for } [GF^n(p^m)] \setminus \{0\}, \quad (3.13)$$

and $|S'| = \frac{p^{mn}-1}{p^m-1}$ by Proposition 3.4. By Theorem 3.10, there exists *1-Generators* S for $GF^n(p^m), |S| = |S'| = \frac{p^{mn}-1}{p^m-1}$. By (3.13) and Theorem 3.10, there does not exist another *1-Generators* S' for $GF^n(p^m), |S'| < |S|$, then

$$S \text{ is a } 1\text{-}Generators \text{ for } GF^n(p^m) \text{ with the smallest number of elements.} \quad (3.14)$$

By Assumption (3.1), $q_{color} = p^m - 1$ and for $k = 1, N = |S| = \frac{p^{mn}-1}{p^m-1}$, we obtain

$$\lfloor log_2 p^{mn} \rfloor = MSDR_1(\frac{p^{mn}-1}{p^m-1}). \quad (3.15)$$

So, by Definition 2.4, Theorem 3.20 and from Lines (3.1), (3.14) and (3.15), there exists the optimal data hiding scheme $(1, \frac{p^{mn}-1}{p^m-1}, \lfloor log_2 p^{mn} \rfloor)$ *for* $q_{color} = p^m - 1$.

Propostion 3.22. *For n is a positive integer, there exists the optimal data hiding scheme* $(1, 2^n - 1, n)$ *for binary, gray and palette images with* $q_{color} = 1$.

Proof. For $q_{color} = 1$, from (3.1), therefore $p = 2, m = 1$. If build a flip graph G, then there exists the optima data hiding scheme $(1, 2^n - 1, n)$ with $q_{color} = 1$ by Theorem 3.21. The Galois field $GF(p^m)$, $GF(p^m) = GF(2)$ is the same as the field Z_2 (see [25]). Next, we show ways to build flip graphs $G = (V, E)$ on the field Z_2 for binary, gray and palette images as follows.

For the binary image, then $C = \{0,1\}$, $c_p \in C$, c_p is a color value of a pixel.

- $V = C$ and for all $v \in V$, the vertex v is assigned a weight by a function *Val* such that $Val(v) = v$.
- $E = \{(c_p, c_{p'}) \mid c_p, c_{p'} \in V, c_p \neq c_{p'}\}$ and every arc $(c_p, c_{p'})$ is assigned the same weight 1.

For the gray image, then $C = \{0,1,...,255\}$, $c_p \in C$, c_p is a color value of a pixel.

- $V = C$ and for all $v \in V$, the vertex v is assigned a weight by a function *Val* such that $Val(v) = v \bmod 2$.

- $E = \{(255,254),(c_p,c_p + 1)|c_p \in V, 1 \le c_p \le 254\}$ and every arc $(c_p, c_{p'})$ is assigned the same weight 1.

For the palette image, then $C = \{0,1,...,2^t - 1\}$, t is the number of bits to represent color indexes, $c_p \in C$, c_p is a color index of a pixel. The palette $P = \{p_0, p_1, \ldots, p_{2^t-1}\}$, $p_i \in P$, p_i is the color corresponding to the color index $i, \forall i = \overline{0, 2^t - 1}$. To unify notations throughout this paper, we change the name of the function *Val* in [14] to Val_p and set $Val(c_p) = Val_p(p)$, where the color index $c_p \in C$ corresponds to the color $p \in P$.

- We consider G to be the rho forest built by FOPA algorithm in [14] and assign the same weight 1 to all arcs of G. However, all colors of the rho forest are replaced with their color indexes.

By Definition 3.12, it is not difficult to verify that the graphs G for binary, gray and palette images built as above are all flip graphs on the field Z_2. So, there exists the optimal data hiding scheme $(1,2^n - 1,n)$ for binary, gray and palette images with $q_{color} = 1$.

Notice that if we set $N = 2^n - 1$, then the data hiding scheme $(1, 2^n - 1, n)$ becomes the data hiding scheme $(1, N, \lfloor log_2(N+1) \rfloor)$. Remember that for N is a positive integer, the data hiding scheme $(1, N, \lfloor log_2(N+1) \rfloor)$ for binary image with $q_{color} = 1$ is the data hiding scheme CTL [7]. So, Proposition 3.22 shows that the data hiding scheme CTL reaches an optimal data hiding scheme for $N = 2^n - 1$, where n is a positive integer.

Theorem 3.23. *Suppose that find a 2-Generators S for the vector space* $GF^n(p^m)$ *with* $|S| = \left\lceil \frac{\frac{p^m - 3}{2} + \sqrt{\frac{(p^m-3)^2}{4} + 2(2^{\lfloor log_2 p^{mn} \rfloor} - 1)}}{p^m - 1} \right\rceil$ *and build a flip graph G. Then there exists the optimal data hiding scheme* $\left(2, |S|, \lfloor log_2 p^{mn} \rfloor\right)$ *for* $q_{color} = p^m - 1$.

Proof. For the assumption of the theorem, by Theorem 3.20, there exists the data hiding scheme $\left(2, |S|, \lfloor log_2 p^{mn} \rfloor\right)$. According to our approach, the data hiding scheme $\left(2, |S|, \lfloor log_2 p^{mn} \rfloor\right)$ is designed based on the assumption $q_{color} = p^m - 1$ by 3.1. Now, we prove it to be optimal for $q_{color} = p^m - 1$.

Suppose the data hiding scheme $(2, N, r)$ is optimal for $q_{color} = p^m - 1$, then

$$r = MSDR_2(N) = \lfloor log_2(1 + q_{color}C_N^1 + q_{color}^2 C_N^2) \rfloor = \lfloor log_2(1 + q_{color}N + q_{color}^2 \frac{N(N-1)}{2}) \rfloor.$$

Therefore

$$2^r \le 1 + q_{color}N + q_{color}^2 \frac{N(N-1)}{2}$$

$$\Leftrightarrow \frac{q_{color}^2}{2}N^2 + q_{color}(1 - \frac{q_{color}}{2})N + 1 - 2^r \ge 0 \quad (3.16)$$

$$\Delta = q_{color}^2(1 - \frac{q_{color}}{2})^2 - 4\frac{q_{color}^2}{2}(1 - 2^r) = q_{color}^2[\frac{(q_{color} - 2)^2}{4} + 2(2^r - 1)]$$

Since $q_{color} \ge 1$, then $\frac{q_{color}^2}{2} > 0$. To have (3.16), we let N hold

$$N \le \frac{\frac{(q_{color}-2)}{2} - \sqrt{\frac{(q_{color}-2)^2}{4} + 2(2^r - 1)}}{q_{color}} \tag{3.17}$$

or

$$N \ge \frac{\frac{(q_{color}-2)}{2} + \sqrt{\frac{(q_{color}-2)^2}{4} + 2(2^r - 1)}}{q_{color}} \tag{3.18}$$

Since $r \ge 1$, then $\frac{\frac{(q_{color}-2)}{2} - \sqrt{\frac{(q_{color}-2)^2}{4} + 2(2^r - 1)}}{q_{color}} < 0$. Since $N \ge 1$, then (3.17) does not hold. Thus N only holds (3.18). By the supposition, the data hiding scheme $(2, N, r)$ is optimal, by Definition (2.4), N is the smallest positive integer and satisfies (3.18), then

$$N = N_{optimum} = \left\lceil \frac{\frac{(q_{color}-2)}{2} + \sqrt{\frac{(q_{color}-2)^2}{4} + 2(2^r - 1)}}{q_{color}} \right\rceil \tag{3.19}$$

For $r = \lfloor log_2 p^{mn} \rfloor$ and $q_{color} = p^m - 1$, from (3.19), we obtain

$$N_{optimum} = \left\lceil \frac{\frac{p^m-3}{2} + \sqrt{\frac{(p^m-3)^2}{4} + 2(2^{\lfloor log_2 p^{mn} \rfloor} - 1)}}{p^m - 1} \right\rceil$$

By the assumption of the theorem, $|S| = \left\lceil \frac{\frac{p^m-3}{2} + \sqrt{\frac{(p^m-3)^2}{4} + 2(2^{\lfloor log_2 p^{mn} \rfloor} - 1)}}{p^m - 1} \right\rceil$, then $|S| =$ $N_{optimum}$. So, the data hiding scheme $(2, \left\lceil \frac{\frac{p^m-3}{2} + \sqrt{\frac{(p^m-3)^2}{4} + 2(2^{\lfloor log_2 p^{mn} \rfloor} - 1)}}{p^m - 1} \right\rceil, \lfloor log_2 p^{mn} \rfloor)$ is optimal for $q_{color} = p^m - 1$. The proof is complete.

Given an image F used as a carrier to embed a secret data sequence into (called a cover image), we partition F into disjoint image blocks of N pixels, $F = \{F_1, F_2, \ldots, F_{t_2}\}$. Let $D = D_1 D_2 \ldots D_{t_3}$ be a secret data sequence embedded in the cover image F, where D_i is secret data of length $\lfloor log_2 p^{mn} \rfloor$ bits, $\forall i = \overline{1, t_3}$. Since each $\lfloor log_2 p^{mn} \rfloor$ bits of secret data is only embedded in one image block of F, $t_3 \le t_2$.

Let *Jump* be a bijective function used to determine the order of blocks in F in the process of hiding D in F, $Jump : \{1,2,...,t_2\} \to \{1,2,...,t_2\}$.

Consider GF to be an arbitrary subset of $2^{\lfloor log_2 p^{mn} \rfloor}$ elements of the set $GF^n(p^m)$, B to be the set of all secret data of length $\lfloor log_2 p^{mn} \rfloor$ bits, it means $B = \{0,1,\ldots,2^{\lfloor log_2 p^{mn} \rfloor} - 1\}$ in the decimal system. Then there exists a bijective function f, $f : B \to GF$.

In real applications, when apply the data hiding scheme $(k, N, \lfloor log_2 p^{mn} \rfloor)$ based on our approach to the process of hiding D in F, we use the secret key set $\mathcal{K}$, $\mathcal{K} = \{K^1, K^2, \ldots, K^{t_1}\}$ instead of one secret key. The process of hiding D in F by using the data hiding scheme $(k, N, \lfloor log_2 p^{mn} \rfloor)$ consists of the embedding algorithm Em_{DF} and the extracting algorithm Ex_{DF} proposed as follows.

The embedding algorithm Em_{DF} (embedding a secret data sequence D in F):

t = 1;
For i = 1 to t_3 Do
{
$M = f(D_i)$; (3.20)
$t = (t - 1) \bmod t_1 + 1$; (3.21)
$F_{Jump(i)} = Em(F_{Jump(i)}, M, K^t)$; // Use the automaton $A(F_{Jump(i)}, M, K^t)$ (3.22)
}
$F' = F$; // F' is called a stego image

The extracting algorithm Ex_{DF} (extracting the secret data sequence D embedded from F'):

$t = 1$;
For $i = 1$ to t_3 Do
{
$t = (t - 1) \bmod t_1 + 1$; (3.23)
$M = Ex(F_{Jump(i)}, K^t)$; // Use the automaton $A(F_{Jump(i)}, M, K^t)$ (3.24)
$D_i = f^{-1}(M)$; // f^{-1} is the inverse function of f (3.25)
}
$D = D_1 D_2 \ldots D_{t_3}$;

Propostion 3.24. *For a cover image F, a secret data sequence D, a bijective function Jump, a bijective function f, a secret key set* $\mathcal{K}$ *and the data hiding scheme* $(k, N, \lfloor log_2 p^{mn} \rfloor)$ *based on our approach given as above. Suppose the stego image block* F' *is generated after D is embedded in F by the embedding algorithm* Em_{DF}*. Then the data sequence* D' *extracted from* F' *by the extracting algorithm* Ex_{DF} *is exactly the secret data sequence D.*

Proof. By (3.21) and (3.23), Em_{DF} in (3.22) and Ex_{DF} in (3.24) use the same secret key K^t. The bijective function *Jump* guarantees $\forall i, j \in \{1,2,\ldots,t_3\}, i \neq j, Jump(i) \neq Jump(j)$, it means that an arbitrary image block in F is only used at most one time in the process of hiding. By Proposition 3.19, M extracted by (3.24) is the same as M embedded by (3.22). Then the bijective function f guarantees that D_i encrypted by (3.20) is the same as D_i decrypted by (3.25), $\forall i \in \{1,2,...,t_3\}$. Therefore we complete the proof.

Security analysis of process of hiding D in F: Assume that we publish parameters k,N, Em, Ex, the vector space $GF^n(p^m)$ and the flip graph G in the data hiding scheme $(k, N, \lfloor log_2 p^{mn} \rfloor)$. The secret element M is extracted from $F'_{Jump(i)}$ by (3.24), we have

$$M = Ex(F'_{Jump(i)}, K^t),$$

from Definitions 3.13 and 3.16 and by (3.9), we obtain

$$M = \sum_{\substack{N \\ i=1 \\ i=1}}^{N} (Val(F'_{Jump(i)}) + K_i^t) v_i. \tag{3.26}$$

From (3.26) and (3.25), to extract accurately $D_i, \forall i = \overline{1, t_3}$, we need to know the *k-Generators* S for $GF^n(p^m)$, the key set K and two bijective functions *Jump*, *f*. Since the number of the *k-Generators* found is $c(p^m - 1)^N$ by Proposition 3.11, then the number of choices for the *k-Generators* S is $c(p^m - 1)^N N!$. The number of choices for the key set $\mathcal{K}$, two bijective functions *Jump* and *f* are $p^{mt_1 N}$, $t_2!$ and $C_{p^{mn}}^{2^{\lfloor log_2 p^{mn} \rfloor}} 2^{\lfloor log_2 p^{mn} \rfloor}!$ (see the security analysis of the data hiding scheme $(k, N, \lfloor log_2 p^{mn} \rfloor)$ as above), respectively. Then for a brute force attack, an attacker has to try every possible combination of S, $\mathcal{K}$, *Jump* and *f* in the given process of hiding. The number of combinations of S, $\mathcal{K}$, *Jump* and *f* is

$$c(p^m - 1)^N N! p^{mt_1 N} t_2! C_{p^{mn}}^{2^{\lfloor log_2 p^{mn} \rfloor}} 2^{\lfloor log_2 p^{mn} \rfloor}!. \tag{3.27}$$

3.3 The Near Optimal and Optimal Data Hiding Schemes for Gray and Palette images

In this subsection, we show that there exist the near optimal data hiding scheme $(2,9,8)$ (Theorem 3.27 and Security analyses (3.45), (3.46)) and the optimal data hiding scheme $(1,5,4)$ (Corollary 3.28 and Security analyses (3.47), (3.48)) for gray and palette images with $q_{color} = 3$.

According to the way of constructing the Galois field $GF(p^m)$ from the polynomial ring $Z_p[x]$, where p is prime and m is a positive integer [25], here consider the case $p = m = 2$ and use the irreducible polynomial $g(x) = x^2 + x + 1$ in $Z_2[x]$ to construct the Galois field $GF(2^2)$ from the polynomial ring $Z_2[x]$, we obtain the Galois field $GF(2^2)$ as follows.

$$GF(2^2) = \{0, 1, x, x+1\}$$

with two operations addition + and multiplication · are defined as in $Z_2[x]$, followed by a reduction modulo $g(x)$.

Notice that the polynomial $g(x)$ is irreducible in $Z_2[x]$. Indeed, if $g(x)$ has factors being different from the constant, then the factors of $g(x)$ are only polynomials of degree 1 and hence $g(x)$ has roots in Z_2, this can not happen because $g(0) = g(1) = 1$.

To save memory space, we write all polynomials of $GF(2^2)$ by sequences of their coefficients and then denote the sequence of any polynomial's coefficients by a binary string and a decimal number as in the following table.

Table 3.1. Elements of the Galois field $GF(2^2)$ represented by binary strings and decimal numbers

Polynomial	Binary string	Decimal number
0	00	0
1	01	1
x	10	2
x + 1	11	3

From **Table 3.1**, to be convenient for programming, hereafter, $GF(2^2)$ can be considered in decimal system by $GF(2^2) = \{0,1,2,3\}$. Then two operations in $GF(2^2)$ are presented as in following table.

Table 3.2. Operations + and · on the Galois field $GF(2^2)$

Operation +

+	0	1	2	3
0	0	1	2	3
1	1	0	3	2
2	2	3	0	1
3	3	2	1	0

Operation ·

·	0	1	2	3
0	0	0	0	0
1	0	1	2	3
2	0	2	3	1
3	0	3	1	2

Based on the binary representation of $GF(2^2)$ as in **Table 3.1**, consider the case $n = 4$, then every vector in the vector space $GF^4(2^2)$ over the field $GF(2^2)$ can be written as a string of length 8 bits, it means that in the decimal system $GF^4(2^2)$ can be presented by $GF^4(2^2) = \{0,1,...,255\}$. Thus two operations of vector addition + and scalar multiplication · on $GF^4(2^2)$ are completely determined based on the operations on the Galois field $GF(2^2)$ in **Table 3.2**.

Example 3.25. Consider the vector space $GF^4(2^2)$ over the field $GF(2^2)$, then 216 + 108 = 180, 2.216 = 108.

Next, we consider the case $k = 2$ and for $p = m = 2$ and $n = 4$ the data hiding scheme $(2, N, 8)$ exists if the hypothesis of Theorem 3.20 holds, it means that find a *2-Generators* S for the vector space $GF^4(2^2)$, $|S| = N$ and build a flip graph G over the Galois field $GF(2^2)$.

By Proposition 3.4, the number of subsets of N elements of the set $[GF^4(2^2)]\setminus\{0\}$ is C_{85}^N. Then to find a *2-[Generators]* S' for the set $[GF^4(2^2)]$, $|S'| = N$, we need to try C_{85}^N subsets of $[GF^4(2^2)]$. This is much more simple than the way to find directly a *2-Generators* S for the vector space $GF^4(2^2)$ with $|S| = N$ because we have to try C_{255}^N subsets of $GF^4(2^2)$.

Based on the representation of the vector space $GF^4(2^2)$ in decimal system, we have created a computer program to find S' and get results that for $N = 8$ there does not exist any S', imply that there does not exist any S, so the data hiding scheme $(2,8,8)$ does not exist by our approach. However, for $N = 9$ the number of S' obtained is $c, c \approx 2^{20}$. Thus by Proposition 3.11, we have approximately $2^{20}.3^9 S$.

Note that the construction of the flip graph over the Galois field $GF(2^2)$ depends on the format of image. In this subsection, we only pay our attention to the gray and palette images. For these image formats, we will point the ways to build flip graphs as follows.

Consider the case of the gray image, then $C = \{0,1,...,255\}$, $c_p \in C$, c_p is a color value of a pixel:

- $V = C$ and for all $v \in V$, the vertex v is assigned a weight by a function *Val* defined by the formula $Val(v) = v \bmod 4$.
- Denote the an arbitrary arc of E by $(c_p, c_{p'})$. Then E and the weights of all arcs of E are given as in the following table.

Table 3.3. The representation of E and the arc weights of G for the gray image

$c_p = 0$

c_p	Weight	$c_{p'}$
0	1	1
	2	2
	3	3

$c_p = 4k, 1 \le k \le 63$

c_p	Weight	$c_{p'}$
k	1	$4k+1$
	2	$4k-2$
	3	$4k-1$

$c_p = 4k+1, 0 \le k \le 63$

c_p	Weight	$c_{p'}$
$4k+1$	1	$4k$
	2	$4k+3$
	3	$4k+2$

$c_p = 4k+2, 0 \le k \le 63$

c_p	Weight	$c_{p'}$
$k+2$	1	$4k+3$
	2	$4k$
	3	$4k+1$

$c_p = 4k+3, 0 \le k \le 62$

c_p	Weight	$c_{p'}$
$4k+3$	1	$4k+2$
	2	$4k+5$
	3	$4k+4$

$c_p = 255$

c_p	Weight	$c_{p'}$
255	1	254
	2	253
	3	252

- It is easy to check that this G is a flip graph over the Galois field $GF(2^2)$.

For the case of the palette image, then $C = \{0,1,\ldots,2^t - 1\}$, t is the number of bits to represent color indexes, $t \ge 2, c_p \in C$, c_p is a color index of a pixel. The palette $P = \{p_0, p_1, \ldots, p_{2^t-1}\}$, $p_i \in P$, p_i is the color corresponding to the color index i, and denote the value of Blue, Green and Red components of the color p_i by $p_i.B$, $p_i.G$ and $p_i.R$, respectively, $\forall i = \overline{0, 2^t - 1}$. For each color index, the set of the closest color indexes is calculated based on the colors in the palette and Euclidean norm (see [9]). The distance between two color indexes i and j, denoted by $d(i,j)$, is determined by the formula

$$d(i,j) = \sqrt{(p_i.B - p_j.B)^2 + (p_i.G - p_j.G)^2 + (p_i.R - p_j.R)^2}.$$

- Let G_C be a directed graph without loops and multiple arcs, $G_C = (V_C, E_C)$, $V_C = C$ and for every vertex $i \in V_C$, i only has three adjacent distinct vertices j_1, j_2, j_3 such that $\forall k = \overline{1,3}, \forall j \in V_C, j \ne i, d(i, j_k) \le d(i, j)$.
- Let Q be a queue data structure to store visited vertices of G_C, operator *Q.Init* initializes $Q = \emptyset$, operator *Q.Count* gets the number of elements contained in Q, operator *Q.Enqueue*(i) adds a element i to the end of Q, operator *Q.Dequeue*() removes and returns a element at the beginning of Q.
- The flip graph $G = (V, E)$ is generated by the breadth first search graph algorithm for G_C (for short, called BFS) given as follows.

```
Procedure BFS(i)
{
    Q.Init;
    Val(i) = 0; // Or another element belonged to GF(2^2)                (3.28)
    pre(i) = −1;
    Q.Enqueue(i);
    While (Q.Count != 0)
    {   i = Q.Dequeue();
        For k = 1 to 3 Do // Take three adjacent vertices of i            (3.29)
            If (Val(j_k) = = −1)                                          (3.30)
            {
                Val(j_k) = Val(i) + k; // The operation addition + on GF(2^2)   (3.31)
                Adjacent(i,k) = j_k;                                      (3.32)
```

```
                    pre(j_k) = i;
                    Q.Enqueue(j_k);
                }
                Else                                                    (3.33)
                    If (Val(j_k) != Val(i) + k)                         (3.34)
                    {
                        If (pre(i) = = −1)                              (3.35)
                        {
                            Choose j ∈ {1,2,3},Val(i) + k = Val(Adjacent(j_k,j));   (3.36)
                            Adjacent(i,k) = Adjacent(j_k,j);            (3.37)
                        }
                        Else                                            (3.38)
                            If (Val(i) + k == Val(pre(i))) Adjacent(i,k) = pre(i);   (3.39)
                            Else                                        (3.40)
                            {
                              Choose j ∈ {1,2,3},Val(i) + k = Val(Adjacent(pre(i),j)); (3.41)
                                Adjacent(i,k) = Adjacent(pre(i),j);     (3.42)
                            }
                    }
                    Else Adjacent(i,k) = j_k;                           (3.43)
            }
        }
    }
    For i ∈ V_C Do Val(i) = −1; // Initialize all the vertices of G_C as not visited
    For i ∈ V_C Do
        If (Val(i) == −1) BFS(i);                                       (3.44)
```

Propostion 3.26. *Let G be the graph generated by BFS for the graph G_C as above. Then G is a flip graph over the Galois field GF*(2^2).

Proof. For $\forall i \in V_C$, i is visited exactly once by BFS, $Val(i)$ is calculated by (3.28) or (3.31), and the operation addition + in (3.31) is an operation on $GF(2^2)$, thus $Val(i) \in GF(2^2)$. Since $V = V_C, Val(i) \in GF(2^2), \forall i \in V$.

Consider $\forall i \in V$, it means that i corresponds to the vertex i of G_C being visited, since $V = V_C = C$, $\forall k \in GF(2^2)\backslash\{0\} = \{1,2,3\}$ corresponds to the adjacent vertex j_k of i in G_C taken by (3.29)). To show that $\exists! j \in V$, the arc (i,j) is assigned k and added to E, we set $j = Adjacent(i,k)$ and prove $Adjacent(i,k)$ to be calculated once such that $Val(Adjacent(i,k)) = Val(i) + k$. Since G_C is traversed by BFS, the pair (i,k) is only used once, then $Adjacent(i,k)$ is calculated at most once. In BFS, if $Adjacent(i,k)$ is calculated, then it holds the condition $Val(Adjacent(i,k)) = Val(i) + k$. So, we only prove that $Adjacent(i,k)$ is always calculated. This means we only show that the vertex j in (3.36) and (3.41) are always found. Suppose $u_1, u_2, \ldots, u_{n_1}$ are all vertices in called order by (3.44), then G_C is traversed by the number of times calling BFS(u), where u in turn equals $u_1, u_2, \ldots, u_{n_1}$. Consider V_C^u to be a set of all visited vertices of G_C by BFS(u). In BFS, the parameter $pre(i) \neq -1$ implies that i is an adjacent vertex of $pre(i)$ in G_C, and $pre(i) = -1$ only happens if $i \in \{u_1, u_2, \ldots, u_{n_1}\}$. Suppose i to be a vertex being visited in G_C, we consider following cases of i:

- In the case $i \in V_C^{u_1}$. Since $t \geq 2$, three adjacent vertices of u_1 hold all (3.30) and then i in (3.35) is always different from u_1, hence $pre(i) \neq -1$, it means that (3.36) is not tested. By BFS, the set of arcs $\{(pre(i),Adjacent(pre[i],k) | k = 1,2,3\}$ is built in G and if (3.40) happens, then we always choose j in (3.41) because $\{Val[pre(i)],Val[Adjacent(pre(i),k)] | k = 1,2,3\} = GF(2^2)$.
- In the case $i \in V_C^{u_{n_2}}, \forall n_2 = \overline{2,n_1}$. If $i = u_{n_2}$ and (3.36) occurs, then G has had the set of arcs $\{(j_k, Adjacent(j_k,k') | k' = 1,2,3\}$ by one of BFS(u_1),..., BFS(u_{n_2-1}), thus find j in (3.36). Conversely, show similarly as the case $i \in V_C^{u_1}$.

So, the graph G generated by BFS given as above is a flip graph. We complete the proof.

Theorem 3.27. *There exists the near optimal data hiding scheme* $(2,9,8)$ *for gray and palette images with* $q_{color} = 3$.

Proof. Based on the Galois field $GF(2^2)$, we find *2-Generators S* for the vector space $GF^4(2^2)$ with $|S| = 9$ and build the flip graphs G for gray and palette images as above, then there exists the data hiding scheme $(2,9,8)$ for gray and palette images with $q_{color} = 3$ by Theorem 3.20 with $k = p = m = 2$, $n = 4$, $N = 9$ and our approach. Next, we show it to be near optimal for $q_{color} = 3$.

Apply Theorem 3.23 with $p = m = 2$, $n = 4$, $q_{color} = 3$, there exists the optima data hiding scheme $(2,8,8)$. For $q_{color} = 3$, apply the Formula (2.1), we have $MSDR_2(9) = \lfloor log_2(1 + 3C_9^1 + 3^2 C_9^2) \rfloor = 8$. So, by Definition 2.5, the data hiding scheme $(2,9,8)$ is near optimal with $q_{color} = 3$. The proof is complete.

Security analysis of the near optimal data hiding scheme $(2,9,8)$: For $p = m = 2$, $n = 4$, $N = 9$, from Formula (3.12), the security of the data hiding scheme (2,9,8) is given by the following formula

$$c3^9 9!2^{18}2^8!. \tag{3.45}$$

Security analysis of the process of hiding D in F by using the data hiding scheme $(2,9,8)$: For $p = m = 2$, $n = 4$, $N = 9$, from Formula (3.27), the security of the process of hiding D in F by using the data hiding scheme $(2,9,8)$ is given by the following formula

$$c3^9 9!2^{18t_1} t_2!2^8!. \tag{3.46}$$

Corollary 3.28. *There exists the optimal data hiding scheme* $(1,5,4)$ *for gray and palette images with* $q_{color} = 3$.

Proof. Based on the Galois field $GF(2^2)$, apply Theorem 3.21 with $p = m = n = 2$, $k = 1$ and use the flip graphs G in the data hiding scheme $(2,9,8)$ for gray and palette images, we obtain immediately the optimal data hiding scheme $(1,5,4)$ for gray and palette images with $q_{colour} = 3$.

Security analysis of the optimal data hiding scheme $(1,5,4)$: By (3.13), $c = 1$. For $p = m = n = 2$, $N = 5$, from Formula (3.12), the security of the data hiding scheme $(1,5,4)$ is given by the following formula

$$3^5 5!2^{10}2^4!. \tag{3.47}$$

Security analysis of the process of hiding D in F by using the data hiding scheme $(1,5,4)$: For $c = 1$, $p = m = n = 2$, $N = 5$, from Formula (3.27), the security of the process of hiding D in F by using the data hiding scheme $(1,5,4)$ is given by the following formula

$$3^5 5! 2^{10t_1} t_2! 2^4!. \tag{3.48}$$

4. Experimental Results

In this section, we make a number of experiments to evaluate efficiency of our proposed data hiding schemes and approach.

Here, we consider the efficiency of a data hiding scheme to be based on two main factors, those are embedding capacity of the cover image and quality of its stego image [15, 18, 29]. Our schemes are the near optimal data hiding scheme $(2,9,8)$ for gray and palette images with q_{color} = 3, and the optimal data hiding scheme $(1, 2^n - 1, n)$ for palette images with q_{color} = 1. To show the efficiency of our approach, we use the same optimal data hiding scheme $(1, N, \lfloor log_2(N+1) \rfloor)$ for binary images with q_{color} = 1, where $N = 2^n - 1$, and carried out comparisons of embedding and extracting time between our and Chang et al.'s approach [7].

All data hiding schemes were implemented in the C# programming language compiled Microsoft Visual Studio 2010 with 64-bit Operating System (Win 7), Intel Core I3, 2.20GHz, 4 GB RAM.

Next, we recall some parameters to determine the efficiency of data hiding schemes.

The total number of the secret data sequence bits embedded in the cover image is called a *Payload*. Corresponding to a certain *Payload*, to measure the embedding capacity of the cover image, we use the embedding rate (*ER*) defined as follows [29].

$$ER = \frac{Payload}{W \times H}(bpp), \tag{4.1}$$

where W and H are the cover image's width and height. Let a given data hiding scheme (k, N, r), denote the maximum of *ER* by ER_{max}, then

$$ER_{\max} = \frac{r}{N}.$$

We use the peak signal to noise ratio (*PSNR*) to evaluate quality of stego image. Based on the value of PSNR, we can know the degree of similarity between the cover image and stego image. If the *PSNR* value is high, then quality of stego image is high. Conversely, quality of stego image is low. In general, for the digital image, *PSNR* is defined by the following formula [8, 15]

$$PSNR = 10\log_{10}\frac{255^2}{MSE}(dB), \tag{4.2}$$

where $MSE = \dfrac{\sum_{i=0}^{W-1}\sum_{i=0}^{H-1}((B(i,j)-B'(i,j))^2 + (G(i,j)-G'(i,j))^2 + (R(i,j)-R'(i,j))^2)}{3 \times W \times H}$, where $B(i,j), G(i,j), R(i,j), B'(i,j), G'(i,j)$ and $R'(i,j)$ are the color value of the Blue, Green and Red components of a pixel at position (i,j) in the cover and stego image, respectively. For human's eyes, the threshold value of *PSNR* value is 30dB [8, 15, 18, 29], it means that the *PSNR* value is higher than 30dB, it is hard to distinguish between the cover image and its stego image.

In order to evaluate the efficiency of our data hiding schemes, we used commonly 8-bit gray and palette cover images given in **Figs. 4.1** and **4.2** to simulate the experiments. **Fig.**

4.3 shows a binary cover image to test the efficiency of our and Chang et al.'s approach. Experimental results are presented in **Tables 4.1, 4.3, 4.2** and **4.4.**

From the **Table 4.1**, we can see that the near optimal data hiding scheme $(2,9,8)$ for gray images with q_{color} = 3 achieves high embedding capacity of cover image (ER = 0.86 bpp) with high stego image quality (the average value of $PSNR$ is 55.84 dB).

As the results show in **Table 4.3**, on average, embedding and extracting time of our approach (BOO) are about 3.38 and 4.24 times faster than that of the Chang et al.'s approach (CTL), respectively.

For the near optimal data hiding scheme $(2,9,8)$ for palette images with q_{color} = 3 and the optimal data hiding scheme $(1,2^n-1,n)$ for palette images with q_{color} = 1, **Tables 4.2** and **4.4** indicate that the average values of $PSNR$ can be higher than the threshold value 30 dB if we choose suitable values of ER.

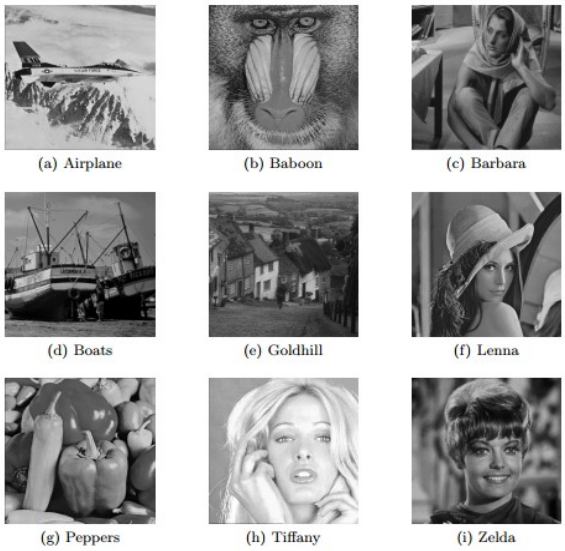

Fig. 4.1. The nine commonly used 8-bit gray cover images sized 512 × 512 pixels

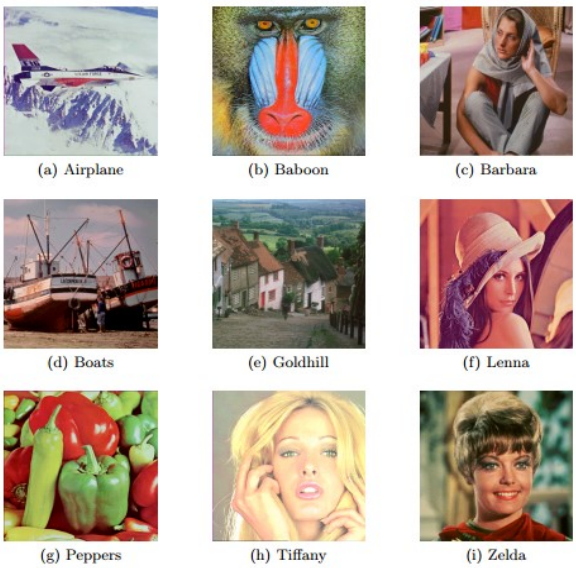

Fig. 4.2. The nine commonly used 8-bit palette cover images sized 512 × 512 pixels

Fig. 4.3. The binary cover image sized 2592 × 1456 pixels

Table 4.1. The payload, *ER* and *PSNR* for the near optimal data hiding scheme $(2,9,8)$ for gray images given in **Fig. 4.1** with $q_{color} = 3$

Image	Airplane	Baboon	Barbara	Boats	Goldhill	Lenna	Peppers	Tiffany	Zelda
ER_{max}	0.89								
Payload	123984								
ER	0.47								
PSNR	61.25	61.11	61.09	59.07	61.00	61.13	61.08	61.00	60.99
Payload	196608								
ER	0.75								
PSNR	57.28	57.12	57.13	55.95	57.03	57.20	57.14	57.08	56.92
Payload	225280								
ER	0.86								
PSNR	56.09	55.94	55.94	55.02	55.86	56.01	55.98	55.88	55.80

Table 4.2. The payload, *ER* and *PSNR* for the near optimal data hiding scheme $(2,9,8)$ for palette images given in **Fig. 4.2** with $q_{color} = 3$

Image	Airplane	Baboon	Barbara	Boats	Goldhill	Lenna	Peppers	Tiffany	Zelda
ER_{max}	0.89								
Payload	5576								
ER	0.02								
PSNR	40.27	38.00	38.42	38.80	38.16	36.16	38.49	31.18	32.94
Payload	9144								
ER	0.03								
PSNR	36.21	33.69	33.66	34.00	33.28	31.25	33.93	26.99	28.09

In the following table, the optimal data hiding schemes $(1,2^n-1,n)$, where $N = 2^n - 1$, for the binary images with $q_{color} = 1$ corresponding to our and Chang et al.'s approach are denoted by BOO and CTL, respectively.

Table 4.3. The comparisons of embedding and extracting time between our and Chang et al.'s approach [7] for the same optimal data hiding scheme $(1, 2^n - 1, n)$, where $N = 2^n - 1$, for the binary image given in **Fig. 4.3** with $q_{color} = 1$. Time is given in second unit.

n	3	4	5	6	7	8
N	7	15	31	63	127	255
ER_{max}	0.43	0.27	0.16	0.1	0.06	0.03
Payload	123984					71328
ER	0.03					0.02
Embedding time (CTL)	0.048	0.055	0.073	0.105	0.166	0.167
Extracting time (CTL)	0.042	0.048	0.064	0.091	0.147	0.154
Embedding time (BOO)	0.010	0.014	0.025	0.035	0.058	0.060
Extracting time (BOO)	0.007	0.010	0.016	0.025	0.042	0.044

Table 4.4. The payload, *ER* and *PSNR* for the optimal data hiding scheme $(1, 2^n - 1, n)$ for palette images given in **Fig. 4.2** with $q_{color} = 1$.

Image		Airplane	Baboon	Barbara	Boats	Goldhill	Lenna	Peppers	Tiffany	Zelda
n = 3	ER_{max}	0.43								
	Payload	2224								
	ER	0.01								
	PSNR	32.72	31.28	31.77	31.45	31.45	30.53	30.67	31.56	31.58
n = 4	ER_{max}	0.27								
	Payload	3200								
	ER	0.01								
	PSNR	31.46	31.21	30.41	30.37	29.99	29.53	29.60	30.45	30.23
n = 5	ER_{max}	0.16								
	Payload	3800								
	ER	0.01								
	PSNR	32.22	31.51	31.17	30.25	30.47	29.76	29.72	31.14	30.13
n = 6	ER_{max}	0.1								
	Payload	3800								
	ER	0.01								
	PSNR	33.49	33.52	32.80	30.80	32.11	31.10	31.17	32.46	31.57
n = 7	ER_{max}	0.06								
	Payload	5576								
	ER	0.02								
	PSNR	31.68	30.27	31.02	28.20	30.21	29.04	29.25	30.41	29.72
n = 8	ER_{max}	0.03								
	Payload	5576								
	ER	0.02								
	PSNR	32.71	31.35	31.79	31.13	30.98	30.12	30.36	31.20	30.80

5. Conclusions

In this paper, we have studied a new approach based on the Galois field $GF(p^m)$ using graph and automata in order to design optimal and near optimal secret data hiding schemes for binary, gray and palette images. By this approach, we showed that the data hiding scheme CTL [7] reaches an optimal data hiding scheme with $N = 2^n - 1$, where n is a positive integer. For the purpose of our work, the proposed schemes consist of the optimal data hiding

scheme $(1,2^n-1,n)$ for binary, gray and palette images with $q_{color}=1$, where n is a positive integer, the near optimal data hiding scheme $(2,9,8)$ for gray and palette images with $q_{color}=3$ and the optimal data hiding scheme $(1,5,4)$ for gray and palette images with $q_{color}=3$. Security analyses indicated that the application of these schemes to the process of hiding a finite sequence of secret data in an image can be prevented from brute-force attacks.

In comparison with Chang et al.'s approach [7], the embedding and extracting time of our approach are about 3.38 and 4.24 times faster than that of theirs, respectively.

Through experimental results, we see that the efficiency (ER = 0.86 bpp, the average value of $PSNR$ is 55.84 dB) of the near optimal data hiding scheme (2,9,8) for gray images with $q_{color}=3$ is indeed better than the efficiency of the HCIH scheme (ER = 0.75 bpp, the average value of $PSNR$ is 46.77 dB) in [29]. We can choose suitable values of ER to achieve acceptable quality of the stego images by applying the near optimal data hiding scheme $(2,9,8)$ for palette images with $q_{color}=3$ and the optimal data hiding scheme $(1,2^n-1,n)$ for palette images with $q_{color}=1$. However, the values of ER is still much lower than $ER_{\max}$. So, the problem of improving the quality of stego images for palette images will be discussed in next work.

An interesting question arises as to whether there exists the optimal data hiding scheme (2,8,8) for 8-bit gray image with $q_{color}=3$.

Acknowledgements

The author is deeply indebted to Phan Trung Huy for supporting this research. The author really would like to thank Phan Thi Ha Duong for her valuable comments and suggestions. This work was partially funded by the Vietnam National Foundation for Science and Technology Development (NAFOSTED) under the grant number 101.99-2016.16.

References

[1] O. I. I. Al-Farraji, "Combination between Steganography and Cryptography in Information Hiding by Using Same Key," *International Journal of Engineering Research and General Science*, 4(6), pp. 201-208, 2016. Article (CrossRef Link).

[2] S. AL-Mansoori, A. Kunhu, "Discrete Cosine Transform and Hash Functions toward Implementing a (Robust-Fragile) Watermarking Scheme," *Proc. SPIE 8895, High-Performance Computing in Remote Sensing III*, 8895, 2013. Article (CrossRef Link).

[3] V. S. Babu, K. J. Helen, "A Study on Combined Cryptography and Steganography," *International, Journal of Research Studies in Computer Science and Engineering*, 2(5), pp. 45-49, 2015. Article (CrossRef Link).

[4] A. Baby, H. Krishnan, "Combined Strength of Steganography and Cryptography - A Literature Survey," *International Journal of Advanced Research in Computer Science*, 8(3), pp. 1007-1010, 2017. Article (CrossRef Link).

[5] G. Birkhoff, S. M. Lane, *A Survey Of Modern Algebra*, Third edition, Macmillan Company, pp. 149-152, 1969.

[6] J. Chen, T. S. Chen, M. W. Cheng, "A New Data Hiding Method in Binary Image," in *Proc. of the IEEE Fifth International Symposium on Multimedia Software Engineering*, pp. 88-93, 2003. Article (CrossRef Link).

[7] C. C. Chang, C. S. Tseng, C. C. Lin, "Hiding Data in Binary Images," in *Proc. of Information Security Practice and Experience, 1st International Conference, ISPEC 2005, Singapore, April 11-14, 2005. Proceedings. LNCS-3439, Springer*, pp. 338-349, 2005. Article (CrossRef Link).

[8] A. Cheddad, J. Condell, K. Curran, P. M. Kevitt, "Digital Image Steganography: Survey and Analysis of Current Methods," *Signal Processing*, 90(3), pp. 727-752, 2010. Article (CrossRef Link).

[9] J. Fridrich, "A New Steganographic Method for Palette-Based Images," in *Proc. of The IS&T PICS Conference*, pp. 285-289, 1999. Article (CrossRef Link).

[10] J. Fridrich, R. Du, "Secure Steganographic Methods for Palette Images," in *Proc. of International Workshop on Information Hiding*, pp. 47-60, 2000. Article (CrossRef Link).

[11] C. L. Hou, C. Lu, S. C. Tsai, W. G. Tzeng, "An Optimal Data Hiding Scheme with TreeBased Parity Check," *IEEE Transactions on Image Processing*, 20(3), pp. 880-886, 2011. Article (CrossRef Link).

[12] P. T. Huy, C. Kim, N. H. Thanh, "Modules over rings of Small Characteristics and Maximality of Data Hiding Ratio in CPTE Schemes," *American Journal of Applied Sciences*, 10(9), pp. 1102-1108, 2013. Article (CrossRef Link).

[13] P. T. Huy, N. H. Thanh, "On the Maximality of Secret Data Ratio in CPTE Schemes," in *Proc. of Intelligent Information and Database Systems,Third International Conference, ACIIDS 2011, Korea, April 20-22, 2011, Proceedings, Part I. LNCS-6591, Springer*, pp. 88-99, 2011. Article (CrossRef Link).

[14] P. T. Huy, N. H. Thanh, N. T. Dat, C. Kim, "Improving Optimal Parity Assignments in Palette Images," *Journal INFORMATION*, 16(4), pp. 2661-2668, 2013.

[15] S. Ijeri, S. Pujeri, B. Shrikant, B. A. Usha, "Image Steganography using Sudoku Puzzle for Secured Data Transmission," *International Journal of Computer Applications*, 48(17), pp. 31-35, 2012. Article (CrossRef Link).

[16] N. Khan, K. S. Gorde, "Data Security by Video Steganography and Cryptography Techniques," *International Journal of Emerging Trends in Electrical and Electronics*, 11(5), pp. 58-64, 2015.

[17] I. Khan, S. Gupta, S. Singh, "A New Data Hiding Approach in Images for Secret Data Communication with Steganography," *International Journal of Computer Applications*, 135(13), pp. 9-14, 2016. Article (CrossRef Link).

[18] W. C. Kuo, S. H. Kuo, C. C. Wang, L. C. Wuu, "High Capacity Data Hiding Scheme Based on Multi-bit Encoding Function," *Optik*, 127(4), pp. 1762-1769, 2016. Article (CrossRef Link).

[19] M. H. Mohamed, L. M. Mohamed, "High Capacity Image Steganography Technique Based on LSB Substitution Method," *Applied Mathematics & Information Sciences*, 10(1), pp. 259-266, 2016. Article (CrossRef Link).

[20] X. Li, S. Cai, W. Zhang, B. Yang, "A Further Study of Large Payloads Matrix Embedding," *Information Sciences*, Vol. 324, pp. 257-269, 2015. Article (CrossRef Link).

[21] H. K. Pan, Y. Y. Chen, Y. C. Tseng, "A Secret of Data Hiding Scheme for Two-Color Images," in *Proc. of ISCC 2000. Fifth IEEE Symposium on Computers and Communications*, pp. 750-755, 2000. Article (CrossRef Link).

[22] K. J. Panchal, F. N. Patel, "Steganography: A Brief Survey," *International Journal of Modern Trends in Engineering and Research*, 2(2), pp. 747-750, 2014. Article (CrossRef Link).

[23] M. I. S. Reddy, A. P. S. Kumar, "Secured Data Transmission Using Wavelet Based Steganography and Cryptography by Using AES Algorithm," *Procedia Computer Science*, Vol. 85, pp. 62-69, 2016. Article (CrossRef Link).

[24] C. A. Stanley, "Pairs of Values and the Chi-squared Attack, Department of Mathematics," *Iowa State University*, May 1, 2005. Article (CrossRef Link).

[25] D. R. Stinson, *Cryptography: Theory and Practice (CRC Press Series on Discrete Mathematics and Its Application)*, CRC Press, pp. 1-20, 180-184, 1995.

[26] Y. C. Tseng, H. K. Pan, "Secure and Invisible Data Hiding in 2-Color Images," in *Proc. of IEEE INFOCOM*, Vol. 2, pp. 887-896, 2001. Article (CrossRef Link).

[27] R. B. Wolfgang, E. J. Delp, "Fragile Watermarking Using The VW2D Watermark," *Proc. SPIE 3657, Security and Watermarking of Multimedia Contents*, vol. 3657, pp. 204-213, 1999. Article (CrossRef Link).
[28] M. Y. Wu, Y. K. Ho, J. H. Lee, "An Iterative Method of Palette Based Image Steganography," *Pattern Recognition Letters*, 25(3), pp. 301-309, 2004. Article (CrossRef Link).
[29] Y. Zhang, J. Jiang, Y. Zha, H. Zhang, S. Zhao, "Research on Embedding Capacity and Efficiency of Information Hiding Based on Digital Images," *International Journal of Intelligence Science*, 3(2), pp. 77-85, 2013. Article (CrossRef Link).

Nguyen Huy Truong was born in Hanoi, Vietnam. He received his Bachelor of Science in Mathematics and Informatics from VNU University of Science in 1999, Bachelor of Economics in Banking from National Economics University in 2000 and Master of Science in Mathematical Assurances for Computers and Computing Systems from VNU University of Science in 2003. He is currently a lecturer in School of Applied Mathematics and Informatics at Hanoi University of Science and Technology. His research interests include theory of codes and applications, information security, quantum information, combinatorial and algebraic structures of some models on graphs.